Studying convergence of gradient algorithms via optimal experimental design theory

Rebecca Haycroft, Luc Pronzato, Henry P. Wynn, Anatoly A. Zhigljavsky

1

Studying convergence of gradient algorithms via optimal experimental design theory

R. Haycroft, L. Pronzato, H.P. Wynn, and A. Zhigljavsky

Summary. We study the family of gradient algorithms for solving quadratic optimization problems, where the step-length γ_k is chosen according to a particular procedure. In order to carry out the study, we re-write the algorithms in a normalized form and make a connection with the theory of optimum experimental design. We provide the results of a numerical study which shows that some of the proposed algorithms are extremely efficient.

1.1 Introduction

In a series of papers (Pronzato *et al.*, 2001, 2002, 2006) and the monograph (Pronzato *et al.*, 2000) certain types of steepest-descent algorithms in $\mathbb{R}^d$ and Hilbert space have been shown to be equivalent to special algorithms for updating measures on the real line. The connection, in outline, is that when steepest-descent algorithms are applied to the minimization of the quadratic function

$$f(x) = \frac{1}{2}(Ax, x) - (x, y) \;, \tag{1.1}$$

where (x, y) is the inner product, they can be translated to the updating of measures in $[m, M]$ where

$$m = \inf_{\|x\|=1} (Ax, x), \qquad M = \sup_{\|x\|=1} (Ax, x)$$

with $0 < m < M < \infty$; m and M are the smallest and largest eigenvalues of A, respectively.

The research has developed from the well known result, due to Akaike (1959) and revisited in (Pronzato *et al.*, 2000; Forsythe, 1968; Nocedal *et al.*, 2002), that for standard steepest descent the renormalized iterates $x_k/\sqrt{\|x_k\|}$ converge to the two-dimensional space spanned by the eigenvectors corresponding to the eigenvalues m and M. Chap. 3 in this volume covers the generalisation to the s-gradient algorithm and draws note on both the author's own work and the important paper of Forsythe (1968).

The use of normalisation is crucial and, in fact, it will turn out that the normalized gradient, rather than the normalized x_k will play the central role.

Thus, let $g(x) = Ax - y$ be the gradient of the objective function (1.1). The steepest-descent algorithm is

$$x_{k+1} = x_k - \frac{(g_k, g_k)}{(Ag_k, g_k)} g_k \ .$$

Using the notation $\gamma_k = (g_k, g_k)/(Ag_k, g_k)$, we write the algorithm as

$$x_{k+1} = x_k - \gamma_k g_k \ .$$

This can be rewritten in terms of the gradients as

$$g_{k+1} = g_k - \gamma_k A g_k \ . \tag{1.2}$$

This is the point at which the algorithms are generalised: we now allow a varied choice of γ_k in (1.2), not necessarily that for steepest descent.

Thus, the main objective of the paper is studying the family of algorithms (1.2) where the step-length γ_k is chosen in a general way described below. To make this study we first rewrite the algorithm (1.2) in a different (normalized) form and then make a connection with the theory of optimum experimental design.

1.2 Renormalized version of gradient algorithms

Let us convert (1.2) to a "renormalized" version. First note that

$$(g_{k+1}, g_{k+1}) = (g_k, g_k) - 2\gamma_k (Ag_k, g_k) + \gamma_k^2 (A^2 g_k, g_k) \ . \tag{1.3}$$

Letting $v_k = (g_{k+1}, g_{k+1})/(g_k, g_k)$ and dividing (1.3) through by (g_k, g_k) gives

$$v_k = 1 - 2\gamma_k \frac{(Ag_k, g_k)}{(g_k, g_k)} + \gamma_k^2 \frac{(A^2 g_k, g_k)}{(g_k, g_k)} \ . \tag{1.4}$$

The value of v_k can be considered as a rate of convergence of algorithm (1.2) at iteration k. Other rates which are asymptotically equivalent to v_k can be considered as well, see Pronzato *et al.* (2000) for a discussion and Theorem 5 in Chap. 3 of this volume. The asymptotic rate of convergence of the gradient algorithm (1.2) can be defined as

$$R = \lim_{k\to\infty} \left(\prod_{i=1}^{k} v_i \right)^{1/k} \ . \tag{1.5}$$

Of course, this rate may depend on the initial point x_0 or, equivalently, on g_0.

To simplify the notation, we need to convert to moments and measures. Since we assume that A is a positive definite d-dimensional square matrix, we can assume, without loss of generality, that A is a diagonal matrix $\Lambda =$

$\mathrm{diag}(\lambda_1, \ldots, \lambda_d)$; the elements $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of the original matrix such that $0 < \lambda_1 \leq \cdots \leq \lambda_d$. Then for any vector $g = (g_{(1)}, \ldots, g_{(d)})^T$ we can define

$$\mu_\alpha(g) = \frac{(A^\alpha g, g)}{(g, g)} = \frac{(\Lambda^\alpha g, g)}{(g, g)} = \frac{\sum_i g_{(i)}^2 \lambda_i^\alpha}{\sum_i g_{(i)}^2} \; .$$

This can be seen as the α-th moment of a distribution with mass $p_i = g_{(i)}^2 / \sum_j g_{(j)}^2$ at λ_i, $i = 1, \ldots, d$. This remark is clearly generalisable to the Hilbert space case.

Using the notation $\mu_\alpha^{(k)} = \mu_\alpha(g_k)$, where g_k are the iterates in (1.2), we can rewrite (1.4) as

$$v_k = 1 - 2\gamma_k \mu_1^{(k)} + \gamma_k^2 \mu_2^{(k)} \; . \tag{1.6}$$

For the steepest-descent algorithm γ_k minimizes $f(x_k - \gamma g_k)$ over γ and we have

$$\gamma_k = \frac{1}{\mu_1^{(k)}} \qquad \text{and} \qquad v_k = \frac{\mu_2^{(k)}}{\mu_1^{(k)^2}} - 1 \; .$$

Write $z_k = g_k / \sqrt{(g_k, g_k)}$ for the normalized gradient and recall that $p_i = g_{(i)}^2 / \sum_j g_{(j)}^2$ is the i-th probability corresponding to a vector g. The corresponding probabilities for the vectors g_k and g_{k+1} are

$$p_i^{(k)} = \frac{(g_k)_{(i)}^2}{(g_k, g_k)} \qquad \text{and} \qquad p_i^{(k+1)} = \frac{(g_{k+1})_{(i)}^2}{(g_{k+1}, g_{k+1})} \qquad \text{for } \; i = 1, \ldots, d \, .$$

Now we are able to write down the re-normalized version of (1.2), which is the updating formula for $p_i \, (i = 1, \ldots, d)$:

$$\begin{aligned} p_i^{(k+1)} &= \frac{(1 - \gamma_k \lambda_i)^2}{(g_k, g_k) - 2\gamma_k (A g_k, g_k) + \gamma_k^2 (A^2 g_k, g_k)} \, p_i^{(k)} \\ &= \frac{(1 - \gamma_k \lambda_i)^2}{1 - 2\gamma_k \mu_1^{(k)} + \gamma_k^2 \mu_2^{(k)}} \, p_i^{(k)} \; . \end{aligned} \tag{1.7}$$

When two eigenvalues of A are equal, say $\lambda_j = \lambda_{j+1}$, the updating rules for $p_j^{(k)}$ and $p_{j+1}^{(k)}$ are identical so that the analysis of the behaviour of the algorithm remains the same when $p_j^{(k)}$ and $p_{j+1}^{(k)}$ are confounded. We may thus assume that all eigenvalues of A are distinct.

1.3 A multiplicative algorithm for optimal design

Optimization in measure spaces covers a variety of areas and optimal experimental design theory is one of them. These areas often introduce algorithms

which typically have two features: the measures are re-weighted in some way and the moments play an important role. Both features arise, as we have seen, in the above algorithms.

In classical optimal design theory for polynomial regression (see, e.g., Fedorov (1972)) one is interested in functionals of the moment (information) matrix $M(\xi)$ of a design measure ξ:

$$M(\xi) = \{m_{ij} : m_{ij} = \mu_{i+j}; 0 \le i, j \le K-1\} ,$$

where $\mu_\alpha = \mu_\alpha(\xi) = \int x^\alpha \mathrm{d}\xi(x)$ are the α-th moments of the measure ξ and K is an integer. For example, when $K = 2$, the case of most interest here, we have

$$M(\xi) = \begin{pmatrix} \mu_0 & \mu_1 \\ \mu_1 & \mu_2 \end{pmatrix} , \tag{1.8}$$

where $\mu_0 = 1$.

Of importance in the theory is the directional (Fréchet) derivative 'towards' a discrete measure ξ_x of mass 1 at a point x. This is

$$\frac{\partial}{\partial\alpha}\Phi\Big(M\big[(1-\alpha)\xi + \alpha\xi_x\big]\Big)\Big|_{\alpha=0} = \mathrm{tr}\Big(\overset{\circ}{\Phi}(\xi)M(\xi_x)\Big) - \mathrm{tr}\Big(\overset{\circ}{\Phi}(\xi)M(\xi)\Big) , \tag{1.9}$$

where

$$\overset{\circ}{\Phi}(\xi) = \frac{\partial\Phi}{\partial M}\Big|_{M=M(\xi)} .$$

Here Φ is a functional on the space of $K \times K$ matrices usually considered as an optimality criterion to be maximized with respect to ξ. The first term on the right hand side of (1.9) is

$$\varphi(x,\xi) = f^T(x)\,\overset{\circ}{\Phi}(\xi)f(x) \tag{1.10}$$

where $f(x) = (1, x, \ldots, x^{K-1})^T$.

A class of optimal design algorithms is based on the multiplicative updating of the weights of the current design measure $\xi^{(k)}$ with some function of $\varphi(x,\xi)$, see Chap. 1 in this volume. We show below how algorithms in this class are related to the gradient algorithms (1.2) in their re-normalized form (1.7).

Assume that our measure is discrete and concentrated on $[m, M]$. Assume also that $\partial\Phi(M)/\partial\mu_{2K-2} > 0$; that is, the (K,K)-element of the matrix $\partial\Phi(M)/\partial M$ is positive. Then $\varphi(x,\xi)$ has a well-defined minimum

$$c(\xi) = \min_{x\in\mathbb{R}} \varphi(x,\xi) > -\infty .$$

Let $\xi(x)$ be the mass at a point x and define the re-weighting at x by

$$\xi'(x) = \frac{\varphi(x,\xi) - c(\xi)}{b(\xi)}\,\xi(x) , \tag{1.11}$$

where $b(\xi)$ is a normalising constant

$$b(\xi) = \int_m^M \big(\varphi(x,\xi) - c(\xi)\big)\xi(\mathrm{d}x) = \int_m^M \varphi(x,\xi)\xi(\mathrm{d}x) - c(\xi)$$
$$= \mathrm{tr}\left[M(\xi)\,\overset{\circ}{\Phi}(\xi)\right] - c(\xi)\,.$$

We see that the first term on the left hand side is (except for the sign) the second term in the directional derivative (1.9). We can also observe that the algorithm (1.11), considered as an algorithm for constructing Φ-optimal designs, belongs to the family of algorithms considered in Chap. 1 of this volume.

We now specialise to the case where $K = 2$. In this case,

$$f(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}\,, \quad \overset{\circ}{\Phi}(\xi) = \begin{pmatrix} \frac{\partial\Phi}{\partial\mu_0} & \frac{1}{2}\frac{\partial\Phi}{\partial\mu_1} \\ \frac{1}{2}\frac{\partial\Phi}{\partial\mu_1} & \frac{\partial\Phi}{\partial\mu_2} \end{pmatrix}$$

and the function $\varphi(x,\xi)$ is quadratic in x:

$$\varphi(x,\xi) = (\,1\,,\,x\,)\begin{pmatrix} \frac{\partial\Phi}{\partial\mu_0} & \frac{1}{2}\frac{\partial\Phi}{\partial\mu_1} \\ \frac{1}{2}\frac{\partial\Phi}{\partial\mu_1} & \frac{\partial\Phi}{\partial\mu_2} \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix} = \frac{\partial\Phi}{\partial\mu_0} + x\frac{\partial\Phi}{\partial\mu_1} + x^2\frac{\partial\Phi}{\partial\mu_2}\,. \quad (1.12)$$

Then

$$c(\xi) = \frac{\partial\Phi}{\partial\mu_0} - B(\xi)\,,$$

where

$$B(\xi) = \frac{1}{4}\left(\frac{\partial\Phi}{\partial\mu_1}\right)^2 \Big/ \left(\frac{\partial\Phi}{\partial\mu_2}\right),$$

and the numerator on the right-hand side of (1.11) is

$$\varphi(x,\xi) - c(\xi) = \frac{\partial\Phi}{\partial\mu_2}\left(x + \frac{1}{2}\frac{\partial\Phi}{\partial\mu_1}\Big/\frac{\partial\Phi}{\partial\mu_2}\right)^2 = B(\xi)\left(1 + 2\frac{\partial\Phi}{\partial\mu_2}\Big/\frac{\partial\Phi}{\partial\mu_1}\,x\right)^2. \quad (1.13)$$

Let us define $\gamma = \gamma(\xi) = \gamma(\mu_1,\mu_2)$ as

$$\gamma = \gamma(\xi) = -2\frac{\partial\Phi}{\partial\mu_2}\Big/\frac{\partial\Phi}{\partial\mu_1}\,. \quad (1.14)$$

We can then write (1.13) as

$$\varphi(x,\xi) - c(\xi) = B(\xi)\,(1 - \gamma(\xi)x)^2\,. \quad (1.15)$$

Normalization is needed to ensure that the measure ξ' is a probability distribution. We obtain that the re-weighting formula (1.11) can be equivalently written as

$$\xi'(x) = \frac{(1-\gamma x)^2}{1 - 2\gamma\mu_1 + \gamma^2\mu_2}\,\xi(x)\,. \quad (1.16)$$

The main, and we consider surprising, connection that we are seeking to make is that this is exactly the same as the general gradient algorithm in its renormalized form (1.7). To see that, we simply write the updating formula (1.16) iteratively

$$\xi^{(k+1)}(x) = \frac{(1-\gamma_k x)^2}{1 - 2\gamma_k \mu_1^{(k)} + \gamma_k^2 \mu_2^{(k)}}\, \xi^{(k)}(x) \; . \tag{1.17}$$

1.4 Constructing optimality criteria which correspond to a given gradient algorithm

Consider now the reverse construction: given some $\gamma = \gamma(\mu_1, \mu_2)$, construct a criterion $\Phi = \Phi(M(\xi))$, where $M(\xi)$ is as in (1.8), which will give this γ in the algorithm (1.17). In general, this is a difficult question.

The dependence on μ_0 is not important here and we can assume that the criterion Φ is a function of the first two moments only: $\Phi(M(\xi)) = \Phi(\mu_1, \mu_2)$.

The relationship between γ and Φ is given by (1.14) and can be written in the form of the following first-order linear partial differential equation:

$$2\frac{\partial \Phi(\mu_1, \mu_2)}{\partial \mu_2} + \gamma(\mu_1, \mu_2)\, \frac{\partial \Phi(\mu_1, \mu_2)}{\partial \mu_1} = 0 \; . \tag{1.18}$$

This equation does not necessarily have a solution for an arbitrary $\gamma = \gamma(\mu_1, \mu_2)$ but does for many particular forms of γ. For example, if $\gamma(\mu_1, \mu_2) = g(\mu_1)h(\mu_2)$ for some functions g and h, then a general solution to the equation (1.18) can be written as

$$\Phi(\mu_1, \mu_2) = F\left(-\int g(\mu_1) \mathrm{d}\mu_1 + 2 \int \frac{1}{h(\mu_2)} \mathrm{d}\mu_2 \right) ,$$

where F is an arbitrary continuously differentiable function such that $\partial \Phi(\mu_1, \mu_2)/\partial \mu_2 > 0$ for all eligible values of (μ_1, μ_2).

Another particular case is $\gamma(\mu_1, \mu_2) = g(\mu_1)\mu_2 + h(\mu_1)\mu_2^{\delta}$ for some functions g and h and a constant δ. Then a general solution to the equation (1.18) is

$$\Phi(\mu_1, \mu_2) = F\left(\mu_2^{1-\delta}\, A_1 + \frac{\delta - 1}{2} \int h\left(\mu_1\right) A_1 \mathrm{d}\mu_1 \right)$$

with $A_1 = \exp\{\frac{1}{2}(\delta - 1) \int g\left(\mu_1\right) \mathrm{d}\mu_1\}$, where F is as above (a C^1-function such that $\partial \Phi(\mu_1, \mu_2)/\partial \mu_2 > 0$).

1.5 Optimum design gives the worst rate of convergence

Let $\Phi = \Phi(M(\xi))$ be an optimality criterion, where $M(\xi)$ is as in (1.8). Associate with it a gradient algorithm with step-length $\gamma(\mu_1, \mu_2)$ as given by (1.14).

Let ξ^* be the optimum design for Φ on $[m, M]$; that is,

$$\Phi(M(\xi^*)) = \max_{\xi} \Phi(M(\xi))$$

where the maximum is taken over all probability measures supported on $[m, M]$. Note that ξ^* is invariant for one iteration of the algorithm (1.16); that is, if $\xi = \xi^*$ in (1.16) then $\xi'(x) = \xi(x)$ for all $x \in \text{supp}(\xi)$.

In accordance with (1.6), the rate associated with the design measure ξ is defined by

$$v(\xi) = 1 - 2\gamma\mu_1 + \gamma^2\mu_2 = \frac{b(\xi)}{B(\xi)} \,. \tag{1.19}$$

Assume that the optimality criterion Φ is such that the optimum design ξ^* is non-degenerate (that is, ξ^* is not just supported at a single point). Note that if $\Phi(M) = -\infty$ for any singular matrix M, then this condition is satisfied.

Since the design ξ^* is optimum, all directional derivatives are non-positive:

$$\frac{\partial}{\partial\alpha}\, \Phi\Big[M\big((1-\alpha)\xi^* + \alpha\xi(x)\big)\Big]\Big|_{\alpha=0^+} \le 0 \,,$$

for all $x \in [m, M]$. Using (1.9), this implies

$$\max_{x\in[m,M]} \varphi(x, \xi^*) \le t^* = \text{tr}\left[M(\xi^*)\, \overset{\circ}{\Phi}\, (\xi^*)\right] \,.$$

Since $\varphi(x, \xi^*)$ is a quadratic convex function of x, this is equivalent to $\varphi(m, \xi^*) \le t^*$ and $\varphi(M, \xi^*) \le t^*$. As

$$\int_m^M \varphi(x, \xi^*)\xi^*(\mathrm{d}x) = t^* \,,$$

this implies that ξ^* is supported at m and M. Since ξ^* is non-degenerate, ξ^* has positive masses at both points m and M and

$$\varphi(m, \xi^*) = \varphi(M, \xi^*) = t^* \,.$$

As $\varphi(x, \xi^*)$ is quadratic in x with its minimum at $1/\gamma$, see (1.15), it implies that

$$\gamma^* = \gamma(\mu_1(\xi^*), \mu_2(\xi^*)) = \frac{2}{m+M} \,.$$

The rate $v(\xi^*)$ is therefore

$$v(\xi^*) = \frac{b(\xi^*)}{B(\xi^*)} = \frac{t^* - c(\xi^*)}{B(\xi^*)} = (1 - m\gamma^*)^2 = (1 - M\gamma^*)^2 = R_{\max} \,,$$

where

$$R_{\max} = \frac{(M-m)^2}{(M+m)^2} \,. \tag{1.20}$$

Assume now that the optimum design ξ^* is degenerate and is supported at a single point x^*. Note that since $\varphi(x, \xi^*)$ is both quadratic and convex, x^* is either m or M. Since the optimum design is invariant in one iteration of the algorithm (1.16), γ^* is constant and

$$\max_{\xi} v(\xi) = \max\left[(1 - m\gamma^*)^2, (1 - M\gamma^*)^2\right] \geq R_{\max}$$

with the inequality replaced by an equality if and only if $\gamma^* = 2/(M + m)$.

1.6 Some special cases

In Table 1.1 we provide a few examples of gradient algorithms (1.2) and indicate the corresponding functions of the probability measures ξ. We only restrict ourselves to the optimality criteria $\Phi(\xi)$ that have the form $\Phi(\xi)$=$\Phi(M(\xi))$, where $M(\xi)$ is the moment matrix (1.8). Neither other moments of ξ nor information about the support of ξ are used for constructing the algorithms below.

For a number of algorithms (most of them have not previously been considered in literature), the table provides the following functions.

- The optimality criterion $\Phi(\xi)$.
- The step-length γ_k in the algorithm (1.2); here γ_k is expressed in the form of $\gamma(\xi)$ as defined in (1.14).
- The rate function $v(\xi)$ as defined in (1.19); this is equivalent to the rate $v_k = (g_{k+1}, g_{k+1})/(g_k, g_k)$ at iteration k for the original algorithm (1.2).
- The φ-function $\varphi(x, \xi)$ as defined in (1.10), see also (1.12).
- The expression for $\mathrm{tr}[M(\xi)\, \overset{\circ}{\Phi}(\xi)]$; this is the quantity that often appears in the right-hand side in the conditions for the optimality of designs.
- The minimum of the φ-function: $c(\xi) = \min_x \varphi(x, \xi)$.

The steepest-descent algorithm corresponds to the case when $\Phi(\xi)$ is the D-optimality criterion. Two forms of this criterion are given in the table; of course, they correspond to the same optimization algorithm. It is well-known that the asymptotic rate of the steepest-descent algorithm is always close to the value $R_{\max}$ defined in (1.20). The asymptotic behaviour of the steepest-descent algorithm has already been extensively studied, see, e.g., Pronzato *et al.* (2000); Akaike (1959); Nocedal *et al.* (2002).

The gradient algorithm with constant step-length $\gamma_k = \gamma$ is well-known in literature. It converges slowly; its rate of convergence can easily be analysed without using the technique of the present paper. We do not study this algorithm and provide its characteristics in the table below only for the sake of completeness.

The steepest-descent algorithm with relaxation is also well-known in literature on optimization. It is known that for suitable values of the relaxation parameter ε this algorithm has a faster convergence rate than the ordinary

Table 1.1. Examples of gradient algorithms

Algorithm	$\Phi(M(\xi))$	$\gamma(\xi)$	$v(\xi)$	$\varphi(x,\xi)=f^T(x)\mathring{\Phi}(\xi)f(x)$	$\mathrm{tr}[M(\xi)\mathring{\Phi}(\xi)]$	$c(\xi)=\min\varphi(x,\xi)$
Steepest Descent (D-optimality)	$\log(\mu_2-\mu_1^2)$ $\mu_2-\mu_1^2$	$\frac{1}{\mu_1}$	$\frac{\mu_2}{\mu_1^2}-1$	$\frac{\mu_2-2x\mu_1+x^2}{\mu_2-\mu_1^2}$ $\mu_2-2x\mu_1+x^2$	2 $2\Phi(\xi)$	1 $\Phi(\xi)$
constant $\gamma_k=\gamma$	$\gamma\mu_2-2\mu_1$	γ	$1-2\gamma\mu_1+\gamma^2\mu_2$	$\gamma\mu_2-2x+\gamma x^2$	$2(\gamma\mu_2-\mu_1)$	$\gamma\mu_2-\frac{1}{\gamma}$
Steepest Descent with relaxation	$\varepsilon\mu_2-\mu_1^2$	$\frac{\varepsilon}{\mu_1}$	$1-2\varepsilon+\varepsilon^2\frac{\mu_2}{\mu_1^2}$	$\mu_2-2x\mu_1+\varepsilon x^2$	$\mu_2(1+\varepsilon)-2\mu_1^2$	$\Phi(\xi)/\varepsilon$
Square-root	$\sqrt{\mu_2}-\mu_1$	$\frac{1}{\sqrt{\mu_2}}$	$2\Big(1-\frac{\mu_1}{\sqrt{\mu_2}}\Big)$	$\frac{\sqrt{\mu_2}}{2}\Big(1-\frac{x}{\sqrt{\mu_2}}\Big)^2$	$\Phi(\xi)$	0
α-root	$\mu_2^\alpha-\mu_1^{2\alpha}$	$\frac{\mu_2^{\alpha-1}}{\mu_1^{2\alpha-1}}$	$\Big(\frac{\mu_2}{\mu_1^2}\Big)^{2\alpha-1}-2\Big(\frac{\mu_2}{\mu_1^2}\Big)^{\alpha-1}+1$	$\alpha(\mu_2^\alpha-2\mu_1^{2\alpha-1}x+\mu_2^{\alpha-1}x^2)$	$2\alpha\Phi(\xi)$	$\alpha\frac{\mu_2^{2\alpha-1}-\mu_1^{4\alpha-2}}{\mu_2^{\alpha-1}}$
α-root with relaxation	$\varepsilon\mu_2^\alpha-\mu_1^{2\alpha}$	$\varepsilon\frac{\mu_2^{\alpha-1}}{\mu_1^{2\alpha-1}}$	$\varepsilon^2\Big(\frac{\mu_2}{\mu_1^2}\Big)^{2\alpha-1}-2\varepsilon\Big(\frac{\mu_2}{\mu_1^2}\Big)^{\alpha-1}+1$	$\alpha(\varepsilon^2\mu_2^\alpha-2\mu_1^{2\alpha-1}x+\varepsilon\mu_2^{\alpha-1}x^2)$	$2\alpha\Phi(\xi)$	$\alpha\frac{\varepsilon^2\mu_2^{2\alpha-1}-\mu_1^{4\alpha-2}}{\varepsilon\mu_2^{\alpha-1}}$
Minimum residues (c-optimality)	$1-\frac{\mu_1^2}{\mu_2}$	$\frac{\mu_1}{\mu_2}$	$1-\frac{\mu_1^2}{\mu_2}$	$\frac{(x\mu_1-\mu_2)^2}{\mu_2^2}$	$\Phi(\xi)$	0
A-optimality	$1/\mathrm{tr}D(\xi)$ $=\frac{\mu_2-\mu_1^2}{1+\mu_2}$	$\frac{(1+\mu_1^2)}{\mu_1(1+\mu_2)}$	$\frac{(1+2\mu_1^2+\mu_1^2\mu_2)(\mu_2-\mu_1^2)}{\mu_1^2(1+\mu_2)^2}$	$\frac{(x-\mu_1)^2+(x\mu_1-\mu_2)^2}{(\mu_2+1)^2}$	$\Phi(\xi)$	$\frac{(\mu_2-\mu_1^2)^2}{(\mu_1^2+1)(\mu_2+1)^2}$

steepest-descent algorithm. However, the reasons why this occurs were not known. In Sect. 1.7, we try to explain this phenomenon. In addition, we prove that if the relaxation parameter is either too small ($\varepsilon < 2m/(m+M)$) or too large ($\varepsilon > 2M/(m+M)$) then the rate of the steepest descent with relaxation becomes worse than $R_{\max}$, the worst-case rate of the standard steepest-descent algorithm.

The square-root algorithm can be considered as a modification of the steepest-descent algorithm. The asymptotic behaviour of this algorithm is now well understood, see Theorem 2 below.

The α-root algorithm is a natural extension of the steepest-descent and the square-root algorithms. The optimality criterion used to construct the α-root algorithm can be considered as the D-optimality criterion applied to the matrix which is obtained from the moment matrix (1.8) by the transformation

$$M(\xi) = \begin{pmatrix} 1 & \mu_1 \\ \mu_1 & \mu_2 \end{pmatrix} \;\to\; M^{(\alpha)}(\xi) = \begin{pmatrix} 1 & \mu_1^\alpha \\ \mu_1^\alpha & \mu_2^\alpha \end{pmatrix} . \tag{1.21}$$

(Of course, other optimality criteria can be applied to the matrix $M^{(\alpha)}(\xi)$, not only the D-criterion.) The asymptotic rate of the α-root algorithm is studied numerically, see Fig. 1.7, Fig. 1.8 and Fig. 1.9. The conclusion is that this algorithm has an extremely fast rate when α is slightly larger than 1.

The α-root algorithm with relaxation (this class of algorithms includes the steepest-descent and square-root algorithms with relaxation) is an obvious generalisation of the algorithm of steepest descent with relaxation. Its asymptotic behaviour is also similar: for a fixed α, for very small and very large values of the relaxation parameter ε the algorithm either diverges or converges with the rate $\geq R_{\max}$. Unless α itself is either too small or too large, there is always a range of values of the relaxation parameter ε for which the rates are much better than $R_{\max}$ and where the algorithm behaves chaotically. The algorithm corresponds to the D-optimality criterion applied to the matrix which is obtained from the original moment matrix (1.8) by the transformation

$$\begin{pmatrix} 1 & \mu_1 \\ \mu_1 & \mu_2 \end{pmatrix} \;\to\; \begin{pmatrix} \varepsilon & \mu_1^\alpha \\ \mu_1^\alpha & \mu_2^\alpha \end{pmatrix} .$$

Of course, this transformation generalizes (1.21).

One can use many different optimality criteria Φ for constructing new gradient algorithms. In Table 1.1 we provide the characteristics of the algorithms associated with two other criteria, both celebrated in the theory of optimal design, namely, c-optimality and A-optimality. The algorithm corresponding to c-optimality, with $\Phi_c(M(\xi)) = (1\,,\ \mu_1)M^{-1}(\xi)(1\,,\ \mu_1)^T$, is the so-called minimum residues optimization algorithm, see Kozjakin and Krasnosel'skii (1982). As shown in (Pronzato *et al.*, 2006), its asymptotic behaviour is equivalent to that of the steepest-descent algorithm (and it is therefore not considered below).

The A-optimality criterion gives rise to a new optimization algorithm for which the expression for the step-length would otherwise have been difficult

to develop. This algorithm is very easy to implement and its asymptotic rate is reasonably fast, see Fig. 1.10.

Many other optimality criteria (and their mixtures) generate gradient algorithms with fast asymptotic rates. This is true, in particular, for the Φ_p-criteria

$$\Phi_p(M(\xi)) = \left(\mathrm{tr} M^{-p}(\xi)\right)^{\frac{1}{p}}$$

for some values of p. For example, a very fast asymptotic rate (see Fig. 1.10) is obtained in the case $p = 2$, where we have

$$\Phi_2(M(\xi)) = 1/\mathrm{tr}\, M^{-2}(\xi) = \frac{(\mu_2 - {\mu_1}^2)^2}{1+{\mu_2}^2+2{\mu_1}^2}\,, \tag{1.22}$$

$$\gamma(\xi) = \frac{1 + 2{\mu_1}^2 + {\mu_1}^2\mu_2}{\mu_1(1+\mu_2+{\mu_1}^2+{\mu_2}^2)}\,,$$

$$\varphi(x,\xi) - c(\xi) = \frac{2(\mu_2-{\mu_1}^2)(x-\mu_1-\mu_1\mu_2-{\mu_2}^2\mu_1+2x{\mu_1}^2+x{\mu_1}^2\mu_2-{\mu_1}^3)^2}{({\mu_1}^2\mu_2 + 2{\mu_1}^2 + 1)({\mu_2}^2 + 2{\mu_1}^2 + 1)^2}\,,$$

and

$$v(\xi) = \frac{(\mu_2 - {\mu_1}^2)({\mu_1}^2{\mu_2}^3 + 2{\mu_1}^2{\mu_2}^2 + 3\mu_2{\mu_1}^2 + 2{\mu_1}^4\mu_2 + 4{\mu_1}^2 + 1 + 3{\mu_1}^4)}{{\mu_1}^2({\mu_2} + {\mu_1}^2 + 1 + {\mu_2}^2)^2}\,.$$

1.7 The steepest-descent algorithm with relaxation

Steepest descent with relaxation is defined as the algorithm (1.2) with

$$\gamma_k = \frac{\varepsilon}{\mu_1}\,, \tag{1.23}$$

where ε is some fixed positive number. The main updating formula (1.7) has the form

$$p_i^{(k+1)} = \frac{(1 - \frac{\varepsilon}{\mu_1}\lambda_i)^2}{1 - 2\varepsilon + \varepsilon^2\frac{\mu_2}{\mu_1^2}}\, p_i^{(k)}\,. \tag{1.24}$$

This can also be represented in the form (1.11) for the design optimality criterion

$$\Phi(M(\xi)) = \varepsilon\mu_2(\xi) - \mu_1^2(\xi)\,. \tag{1.25}$$

The rate $v(\xi)$ associated with a design ξ was defined in (1.19) and is simplified to

$$v(\xi) = 1 - 2\,\varepsilon + \varepsilon^2\mu_2/\mu_1^2\,.$$

The following lemma reveals properties of the optimum designs for the optimality criteria (1.25).

Lemma 1. *Let $0 < m < M$, $\varepsilon > 0$ and let ξ^* be the optimum design corresponding to the optimality criterion* (1.25)*. Then ξ^* is supported at two points m and M with the weight $\xi^*(m)$ as in Table 1.2 and $\xi^*(M) = 1 - \xi^*(m)$.*

Table 1.2. Values of $\xi^*(m)$, $\Phi(M(\xi^*))$ and $v(\xi^*)$

	$0 < \varepsilon \le \frac{2m}{m+M}$	$\frac{2m}{m+M} \le \varepsilon \le \frac{2M}{m+M}$	$\varepsilon \ge \frac{2M}{m+M}$
$\xi^*(m)$	1	$\frac{2M-\varepsilon(M+m)}{2(M-m)}$	0
$\Phi(M(\xi^*))$	$m^2(\varepsilon-1)$	$\frac{1}{4}\varepsilon^2(m+M)^2-\varepsilon mM$	$M^2(\varepsilon-1)$
$v(\xi^*)$	$(1-\varepsilon)^2$	$R_{\max}$	$(\varepsilon-1)^2$

The proof is straightforward.

In addition to the values of the weights $\xi^*(m)$, Table 1.2 contains the values of the optimality criterion $\Phi(M(\xi))$ and the rate function $v(\xi)$ for the optimum design $\xi = \xi^*$.

Theorem 1 below shows that if the relaxation coefficient ε is either small ($\varepsilon < 4Mm/(M+m)^2$) or large ($\varepsilon > 1$), then for almost all starting points the algorithm asymptotically behaves as if it has started at the worst possible initial point. However, for some values of ε the rate does not attract to a constant value and often exhibits chaotic behaviour. Typical behaviour of the asymptotic rate (1.5) is shown in Fig. 1.1 where we display the asymptotic rates in the case $M/m = 10$.

In this figure and all other figures in this chapter we assume that $d = 100$ and all the eigenvalues are equally spaced. We have established numerically that the dependence on the dimension d is insignificant as long as $d \ge 10$. In particular, we found that there is virtually no difference in the values of the asymptotic rates corresponding to the cases $d = 100$ and $d = 10^6$, for all the algorithms studied. In addition, choosing equally spaced eigenvalues is effectively the same as choosing eigenvalues uniformly distributed on $[m, M]$ and taking expected values of the asymptotic rates.

Theorem 1. *Assume that ε is such that either $0 < \varepsilon < 4Mm/(m+M)^2$ or $\varepsilon > 1$. Let ξ_0 be any non-degenerate probability measure with support $\{\lambda_1, \dots, \lambda_d\}$ and let the sequence of probability measures $\{\xi^{(k)}\}$ be defined via the updating formula (1.24) where $p_i^{(k)}$ are the masses $\xi^{(k)}(\lambda_i)$. Then the following statements hold:*

- *for any starting point x_0, the sequence $\Phi_k = \Phi(M(\xi^{(k)}))$ monotonously increases ($\Phi_0 \le \Phi_1 \le \cdots \le \Phi_k \le \cdots$) and converges to a limit $\lim_{k\to\infty} \Phi_k$.*
- *For almost all starting points x_0 (with respect to the uniform distribution of $g_0/\sqrt{||g_0||}$ on the unit sphere in $\mathbb{R}^d$),*
  – the limit $\lim_{k\to\infty} \Phi_k$ does not depend on the initial measure $\xi^{(0)}$ and is

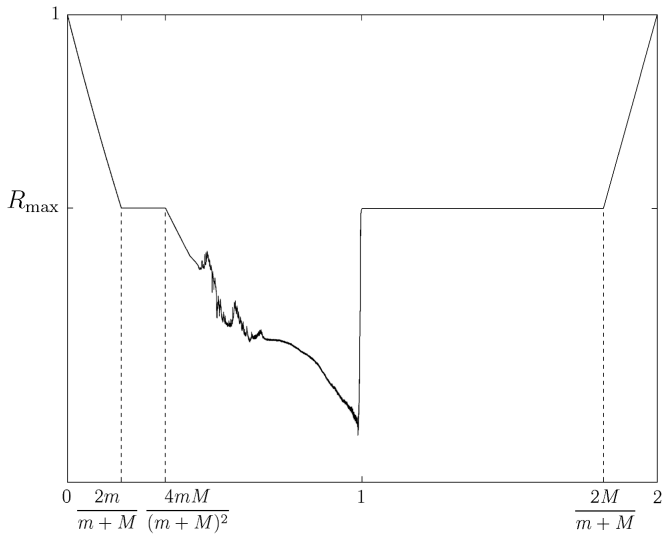

Fig. 1.1. Asymptotic rate of convergence as a function of ε for the steepest-descent algorithm with relaxation ε

equal to $\Phi(M(\xi^*))$ *as defined in Table 1.2;*
– the sequence of probability measures $\{\xi^{(k)}\}$ *converges* (*as* $k \to \infty$) *to the optimum design* ξ^* *defined in Lemma 1, and*
– the asymptotic rate R *of the steepest-descent algorithm with relaxation, as defined in* (1.2) *and* (1.23)*, is equal to* $v(\xi^*)$ *defined in Table 1.2.*

Proof. Note that $\Phi_k \le 0$ for all $\xi^{(k)}$ if

$$0 < \varepsilon < \min_{\xi} \frac{\mu_1^2(\xi)}{\mu_2(\xi)} = \frac{4Mm}{(m+M)^2}\,.$$

If $\varepsilon \ge 1$ then $\Phi_k \ge 0$ for any $\xi^{(k)}$ in view of the Cauchy–Schwarz inequality. Additionally, $|\Phi_k| \le (1+\varepsilon)M^2$ for all $\xi^{(k)}$ so that $\{\Phi_k\}$ is always a bounded sequence.

The updating formulae for the moments are

$$\mu_1' = (\mu_1^3 - 2\varepsilon\mu_1\mu_2 + \varepsilon^2\mu_3)/W \quad \text{and} \quad \mu_2' = (\mu_1^2\mu_2 - 2\varepsilon\mu_1\mu_3 + \varepsilon^2\mu_4)/W$$

where $W = \varepsilon^2\mu_2 + \mu_1^2(1-2\varepsilon)$ and

$$\mu_\alpha = \mu_\alpha(\xi^{(k)}),\ \ \mu_\alpha' = \mu_\alpha(\xi^{(k+1)}),\ \forall\alpha,\ k = 0, 1, \ldots \tag{1.26}$$

The sequence Φ_k is non-decreasing if $\Phi_{k+1} - \Phi_k \ge 0$ holds for any probability measure $\xi = \xi^{(k)}$. If for some k the measure $\xi^{(k)}$ is degenerate (that is, has mass 1 at one point) then $\xi^{(k+1)} = \xi^{(k)}$ and the first statement of the

theorem holds. Let us suppose that $\xi = \xi^{(k)}$ is non-degenerate for all k. In particular, this implies $\mu_2 > \mu_1^2$ and

$$W = \varepsilon^2\mu_2 + \mu_1^2(1-2\varepsilon) = \mu_1^2(1-\varepsilon)^2 + \varepsilon^2(\mu_2 - \mu_1^2) > 0\,.$$

We have:

$$\Phi_{k+1} - \Phi_k \geq 0 \Longleftrightarrow \left(\varepsilon\mu_2^{'} - (\mu_1^{'})^2\right) - \left(\varepsilon\mu_2 - \mu_1^2\right) \geq 0\,. \qquad (1.27)$$

We can represent the left-hand side of the second inequality in (1.27) as

$$\left(\varepsilon\mu_2^{'} - (\mu_1^{'})^2\right) - \left(\varepsilon\mu_2 - \mu_1^2\right) = \varepsilon\frac{U}{W^2}\,,$$

where

$$\begin{aligned} U = {} & W(\mu_1^2\mu_2 - 2\varepsilon\mu_1\mu_3 + \varepsilon^2\mu_4 - W\mu_2) \\ & + (W\mu_1 + \mu_1^3 - 2\varepsilon\mu_1\mu_2 + \varepsilon^2\mu_3)(\varepsilon\mu_1\mu_2 - \varepsilon\mu_3 - 2\,{\mu_1}^3 + 2\,\mu_1\mu_2)\,. \end{aligned}$$

As W^2 will always remain strictly positive, the problem is reduced to determining whether or not $U \geq 0$. To establish the inequality $U \geq 0$, we show that $U = V(a,b)$ for some a and b, where

$$V(a,b) = \mathrm{var}(aX + bX^2) = a^2\mu_2 + 2ab\mu_3 + b^2\mu_4 - (a\mu_1 + b\mu_2)^2 \geq 0 \quad (1.28)$$

and X is the random variable with distribution ξ.

Consider $U - V(a,b) = 0$ as an equation with respect to a and b and prove that there is a solution to this equation. First, choose b to eliminate the μ_4 term: $b = b_0 = \varepsilon\sqrt{W}$. The value of b_0 is correctly defined as $W > 0$.

The next step is to prove that there is a solution to the equation $U - V(a,b_0) = 0$ with respect to a. Note that $U - V(a,b_0)$ is a quadratic function in a. Let D be the discriminant of this quadratic function; it can be simplified to

$$D = (\varepsilon - 1)\left(\varepsilon\mu_2 - {\mu_1}^2\right)\left(\mu_3\varepsilon + 2\,{\mu_1}^3 - \varepsilon\mu_1\mu_2 - 2\,\mu_2\mu_1\right)^2\,.$$

This is clearly non-negative for $\varepsilon > 1$ and $0 < \varepsilon < 4Mm/(m+M)^2$. Therefore, there exist some a and b such that $U = V(a,b)$. This implies $\Phi_{k+1} - \Phi_k \geq 0$ and therefore $\{\Phi_k\}$ is a monotonously increasing bounded sequence converging to some limit $\Phi_* = \lim_{k\to\infty}\Phi_k$.

Consider now the second part of the theorem. Assume that the initial measure ξ_0 is such that $\xi_0(m) > 0$ and $\xi_0(M) > 0$.

From the sequence of measures $\{\xi^{(k)}\}$ choose a subsequence weakly converging to some measure ξ_* (we can always find such a sequence as all the measures are supported on an interval). Let $X = X_*$ be the random variable defined by the probability measure ξ_*. For this measure, the value of V defined in (1.28) is zero as $\Phi(M(\xi^{(k+1)})) = \Phi(M(\xi^{(k)}))$ if $\xi^{(k)} = \xi_*$. Therefore, the random variable $aX_* + bX_*^2$ is degenerated (here a and b are some coefficients)

and therefore X_* is concentrated at either a single point or at two distinct points.

Assume that $\varepsilon < 1$ and therefore $0 < \varepsilon < 4Mm/(m+M)^2$ (similar arguments work in the case $\varepsilon > 1$). Then, similar to the proof for the steepest-descent algorithm (see (Pronzato *et al.*, 2000, p. 175)) one can see that the masses $\xi^{(k)}(m)$ are bounded away from 0; that is, $\xi^{(k)}(m) \geq c$ for some $c > 0$ and all k, as long as $\xi_0(m) > 0$. Therefore, the point m always belongs to the support of the measure ξ_*. If $0 < \varepsilon < 2m/(m+M)$ then one can easily check that ξ_* must be concentrated at a single point (otherwise $U = U(\xi_*) \neq 0$); this point is necessarily m and therefore the limiting design ξ_* coincides with the optimal design ξ^*. If $2m/(m+M) < \varepsilon < 4Mm/(m+M)^2$, then using similar arguments one can find that the second support point of ξ_* is M for almost all starting points.[1] As long as the support of ξ_* is established, one can easily see that the only two-point design that has $U(\xi_*) = 0$ is the optimal design ξ^* with respect to the criterion (1.25). ■

One of the implications of the theorem is that if the relaxation parameter is either too small ($\varepsilon < 2m/(m+M)$) or too large ($\varepsilon > 2M/(m+M)$) then the rate of the steepest-descent algorithm with relaxation becomes worse than $R_{\max}$, the worst-case rate of the standard steepest-descent algorithm. As a consequence, we also obtain a well-known result that if the value of the relaxation coefficient is either $\varepsilon < 0$ or $\varepsilon > 2$, then the steepest descent with relaxation diverges. When $4Mm/(m+M)^2 < \varepsilon \leq 1$ the relaxed steepest-descent algorithm does not necessarily converge to the optimum design. It is within this range of ε that improved asymptotic rates of convergence are demonstrated, see Fig. 1.1.

The convergence rate of all gradient-type algorithms depends on, amongst other things, the condition number $\varrho = M/m$. As one would expect, an increase in ϱ gives rise to a worsened rate of convergence. The improvement yielded by the addition of a suitable relaxation coefficient to the steepest-descent algorithm however, produces significantly better asymptotic rates of convergence than standard steepest descent. Fig. 1.2 shows the effect of increasing the value of ϱ on the rates of convergence for the steepest-descent algorithm with relaxation coefficients $\varepsilon = 0.97$ and $\varepsilon = 0.99$ (see also Fig. 1.10). For large d and ϱ, we expect that the asymptotic convergence rates for this family of algorithms will be bounded above by $R_{\max}$ and below by $R_{\min}$ where

$$R_{\min} = \left(\frac{\sqrt{M}-\sqrt{m}}{\sqrt{M}+\sqrt{m}}\right)^2 = \left(\frac{\sqrt{\varrho}-1}{\sqrt{\varrho}+1}\right)^2; \tag{1.29}$$

[1] The only possibility for M to vanish from the support of the limiting design ξ_* is to obtain $\mu_1(\xi^{(k)})/\varepsilon = M$ at some iteration k, but this obviously almost never (with respect to the distribution of $g_0/\sqrt{||g_0||}$) happens. Note that in this case M will be replaced with λ_{d-i} for some $i \geq 1$ which can only improve the asymptotic rate of convergence of the optimization algorithm.

the rate $R_{\min}$ is exactly the same as the rate N^*_∞ defined in (3.22), Chap. 3. The relaxation coefficient $\varepsilon = 0.99$ produces asymptotic rates approaching $R_{\min}$, see also Fig. 1.10).

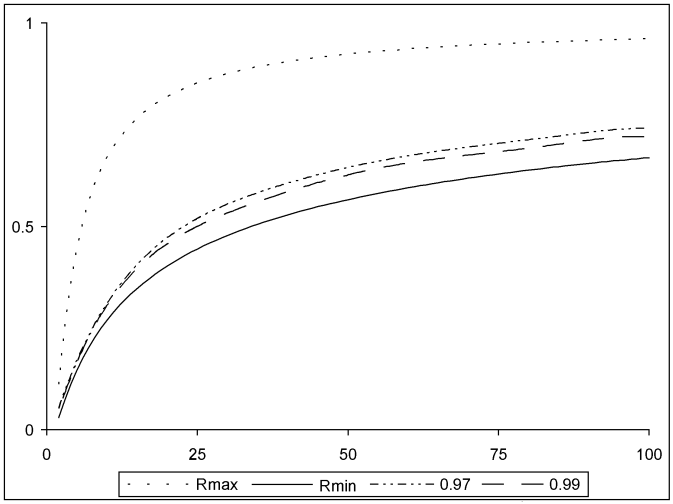

Fig. 1.2. Asymptotic rate of convergence as a function of ϱ for steepest descent with relaxation coefficients $\varepsilon = 0.97$ and $\varepsilon = 0.99$

In Fig. 1.3 we display 250 rates $v(\xi_k)$, $750 < k \leq 1000$ which the steepest-descent algorithms with relaxation coefficients $\varepsilon \in [0.4, 1]$ attained starting at the random initial designs ξ_0. In this figure, we observe bifurcations and transitions to chaotic regimes occurring for certain values of ε.

Fig. 1.4 shows the same results but in the form of the log-rates, $-\log(v(\xi_k))$. Using the log-rates rather than the original rates helps to see the variety of small values of the rates, which is very important as small rates $v(\xi_k)$ force the final asymptotic rate to be small.

Fig. 1.5 shows the log-rates, $-\log(v(\xi_k))$, occurring for $\varepsilon \in [0.99, 1.0]$. This figure illustrates the effect of bifurcation to chaos when we decrease the values of ε starting at 1.

1.8 Square-root algorithm

For the square-root algorithm, we have:

$$\Phi(M(\xi)) = \sqrt{\mu_2} - \mu_1, \quad v = v(\xi) = 2(1 - \frac{\mu_1}{\sqrt{\mu_2}}) \, .$$

In the present case, the optimum design ξ^* is concentrated at the points m and M with weights

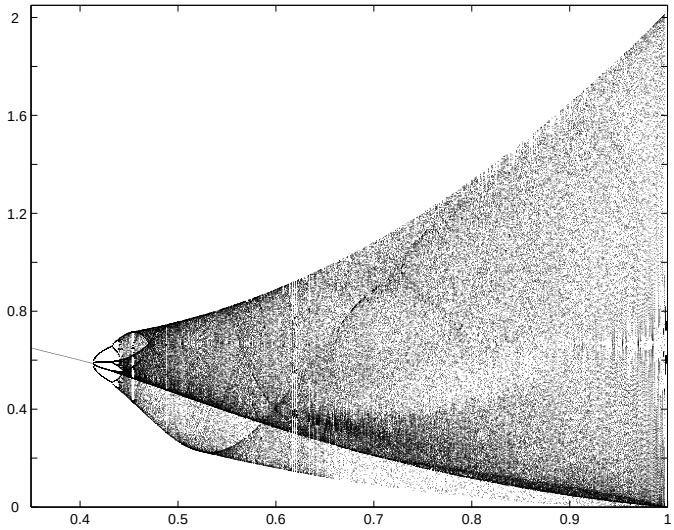

Fig. 1.3. Rates $v(\xi_k)$ $(750 < k \leq 1000)$ for steepest descent with relaxation; varying ε, $\varrho = 10$

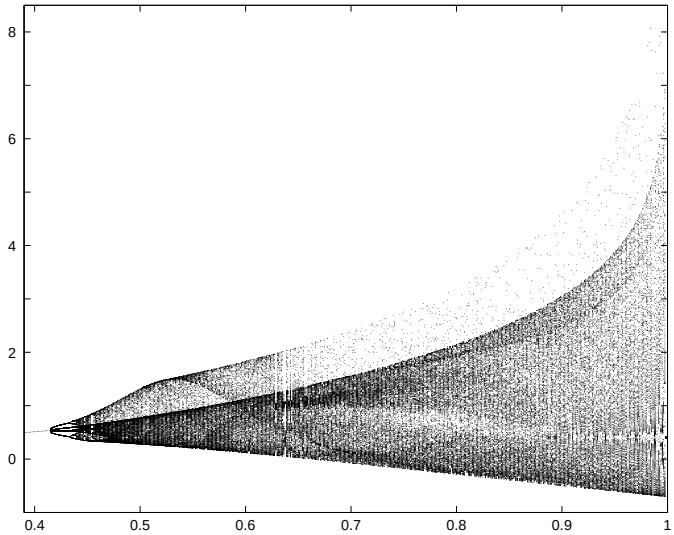

Fig. 1.4. Log-rates $-\log(v(\xi_k))$ $(750 < k \leq 1000)$ for steepest descent with relaxation; varying ε, $\varrho = 10$

$$\xi^*(M) = \frac{3m + M}{4(m + M)}, \quad \xi^*(m) = \frac{m + 3M}{4(m + M)}\,. \tag{1.30}$$

For the optimum design, we have

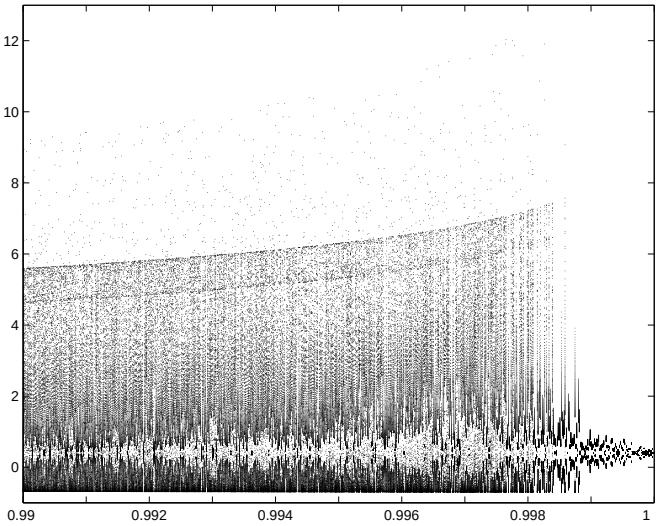

Fig. 1.5. Log-rates $-\log(v(\xi_k))$ $(750 < k \leq 1000)$ for steepest descent with relaxation; $\varrho = 10$, $\varepsilon \in [0.99, 1.0]$

$$\Phi(M(\xi^*)) = \frac{1}{4}\frac{(M-m)^2}{m+M} \quad \text{and} \quad v(\xi^*) = R_{\max}\,.$$

The main updating formula (1.7) has the form

$$p_i^{(k+1)} = \frac{(1-\frac{1}{\sqrt{\mu_2}}\lambda_i)^2}{2(1-\frac{\mu_1}{\sqrt{\mu_2}})}\, p_i^{(k)}\,. \tag{1.31}$$

Theorem 2. *Let ξ_0 be any non-degenerate probability measure with support $\{\lambda_1, \ldots, \lambda_d\}$ and let the sequence of probability measures $\{\xi^{(k)}\}$ be defined via the updating formula* (1.31) *where $p_i^{(k)}$ are the masses $\xi^{(k)}(\lambda_i)$. Then the following statements hold:*

- *for any starting point x_0, the sequence $\Phi_k = \Phi(M(\xi^{(k)}))$ monotonously increases $(\Phi_0 \leq \Phi_1 \leq \cdots \leq \Phi_k \leq \cdots)$ and converges to a limit $\lim_{k\to\infty} \Phi_k$.*
- *If the starting point x_0 of the optimization algorithm is such that $\xi^{(0)}(\lambda_1) > 0$ and $\xi^{(0)}(\lambda_d) > 0$, then*
  – the limit $\lim_{k\to\infty} \Phi_k$ does not depend on the initial measure $\xi^{(0)}$ and is equal to $\Phi(M(\xi^*)) = (M-m)^2/[4(m+M)]$;
  – the sequence of probability measures $\{\xi^{(k)}\}$ converges (as $k \to \infty$) to the optimum design ξ^* defined in (1.30)*, and*
  – the asymptotic rate R of the square-root optimization algorithm is equal to $R_{\max}$.

Proof. The proof is similar to (and simpler than) the proof of Theorem 1.

For the square-root algorithm, we have

$$\mu_1' = \left(\mu_1 - 2\sqrt{\mu_2} + \frac{\mu_3}{\mu_2}\right)/v\,, \quad \mu_2' = \left(\mu_2 - 2\,\frac{\mu_3}{\sqrt{\mu_2}} + \frac{\mu_4}{\mu_2}\right)/v\,,$$

where we use the notation for the moments introduced in (1.26).

Furthermore,

$$\Phi_{k+1} - \Phi_k = \sqrt{\mu_2'} - \mu_1' - \sqrt{\mu_2} + \mu_1\,,$$

$$\Phi_{k+1} \geq \Phi_k \Longleftrightarrow \sqrt{\mu_2'} \geq \mu_1' + \sqrt{\mu_2} - \mu_1 = \frac{\mu_3 - \mu_1\mu_2}{2\sqrt{\mu_2}\left(\sqrt{\mu_2} - \mu_1\right)} - \mu_1\,.$$

The inequality

$$\mu_2' \geq (\mu_1' + \sqrt{\mu_2} - \mu_1)^2 \tag{1.32}$$

would therefore imply $\Phi_{k+1} \geq \Phi_k$ for any design $\xi = \xi^{(k)}$.

Let X be a random variable with probability distribution $\xi = \xi_k$. Then we can see that the difference $\mu_2' - (\mu_1' + \sqrt{\mu_2} - \mu_1)^2$ can be represented as

$$\mu_2' - (\mu_1' + \sqrt{\mu_2} - \mu_1)^2 = \frac{1}{2\sqrt{\mu_2}(\sqrt{\mu_2} - \mu_1)}\mathrm{var}(aX + X^2) \geq 0$$

where

$$a = \kappa + \frac{\kappa + 2\sqrt{\mu_2}}{\sqrt{2}}\sqrt{1 - \frac{\mu_1}{\sqrt{\mu_2}}} \text{ and } \kappa = \frac{\mu_1\mu_2 - \mu_3}{\mu_2 - \mu_1^2}\,.$$

This implies (1.32) and therefore the monotonic convergence of the sequence $\{\Phi_k\}$.

As a consequence, any limiting design for the sequence $\{\xi_k\}$ is concentrated at two points. If $\xi^{(0)}(\lambda_1) > 0$ and $\xi^{(0)}(\lambda_d) > 0$, then there is a constant c_0 such that $\xi^{(k)}(\lambda_1) > c_0$ and $\xi^{(k)}(\lambda_d) > c_0$ for all k implying that the limiting design is concentrated at m and M. The only design with support $\{m, M\}$ that leaves the value of $\Phi(M(\xi))$ unchanged is the optimal design ξ^* with weights (1.30). The rate $v(\xi^*)$ for this algorithm is $R_{\max}$. ■

1.9 *A*-optimality

Consider the behaviour of the gradient algorithm generated by the A-optimality criterion in the two-dimensional case; that is, when $d = 2$, $\lambda_1 = m$ and $\lambda_2 = M$. Assume that the initial point x_0 is such that $0 < \xi^{(0)}(m) < 1$ (otherwise the initial design $\xi^{(0)}$ is degenerated and so are all other designs $\xi^{(k)}$, $k \geq 1$).

Denote $p_k = \xi^{(k)}(m)$ for $k = 0, 1, \ldots$. As $d = 2$, all the designs $\xi^{(k)}$ are fully described by the corresponding values of p_k. In the case of A-optimality, the updating formula for the p_k's is $p_{k+1} = f(p_k)$ where

$$f(p) = \left(1 - \frac{\left(1+{\mu_1}^2\right)}{\mu_1\,(1+\mu_2)}m\right)^2 \frac{{\mu_1}^2\,(1+\mu_2)^2}{(1+2\,{\mu_1}^2+{\mu_1}^2\mu_2)\,(\mu_2-{\mu_1}^2)}\,p\,,$$

$\mu_1 = pm + (1-p)M$ and $\mu_2 = pm^2 + (1-p)M^2$. The fixed point of the transformation $p_{k+1} = f(p_k)$ is

$$p_* = \frac{M^2+1-\sqrt{(M^2+1)(m^2+1)}}{M^2-m^2}\,.$$

For this point we have $p_* = f(p_*)$ and the design with the mass p_* at m and mass $1-p_*$ at M is the A-optimum design for the linear regression model $y_j = \theta_0 + \theta_1 x_j + \varepsilon_j$ on the interval $[m, M]$ (and any subset of this interval that includes m and M). This fixed point p_* is unstable for the mapping $p \to f(p)$ as $|f'(p_*)| > 1$.

For the transformation $f^2(\cdot) = f(f(\cdot))$, see Fig. 1.6 for an illustration of this map, there are two stable fixed points which are 0 and 1. The fact that the points 0 and 1 are stable for the mapping $p \to f^2(p)$ follows from

$$(f(f(p)))'\Big|_{p=0} = (f(f(p)))'\Big|_{p=1} = f'(0)f'(1) = \frac{(Mm+1)^4}{(m^2+1)^2\,(M^2+1)^2}\,;$$

the right-hand side of this formula is always positive and less than 1. There is a third fixed point for the mapping $p \to f^2(p)$; this is of course p_* which is clearly unstable.

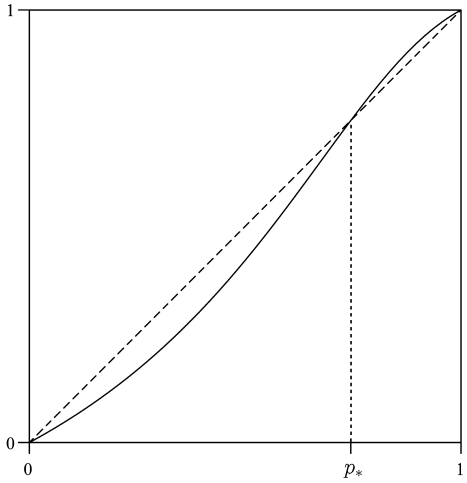

Fig. 1.6. Graph depicting the transformation $f^2(\cdot)$ for $m = 1$, $M = 4$

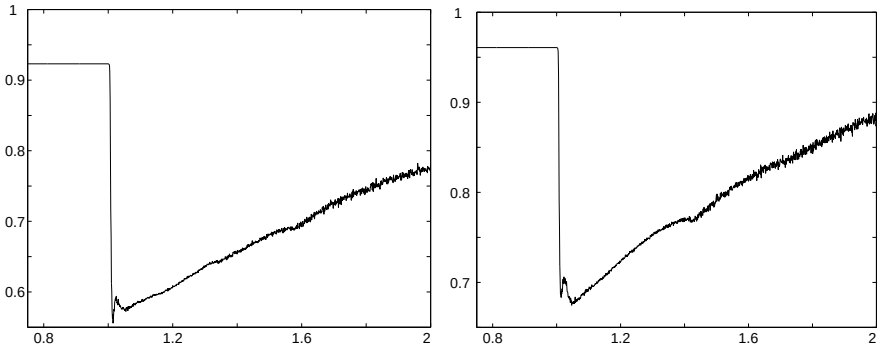

Fig. 1.7. Asymptotic rates for the α-root gradient algorithm; left: $\varrho = 50$, right: $\varrho = 100$

This implies that in the two-dimensional case the sequence of measures $\xi^{(k)}$ attracts (as $k \to \infty$) to a cycle of oscillations between two degenerate measures, one is concentrated at m and the other one is concentrated at M. The superlinear convergence of the corresponding gradient algorithm follows from the fact that the rates $v(\xi)$ at these two degenerate measures are 0 (implying $v_k \to 0$ as $k \to \infty$ for the sequence of rates v_k).

To summarize the result, we formulate it as a theorem.

Theorem 3. *For almost all starting points x_0, the gradient algorithm corresponding to the A-optimality criterion for $d = 2$ has a superlinear convergence in the sense that the sequence of rates v_k tends to 0 as $k \to \infty$.*

If the dimension d is larger than 2, then the convergence of the optimization algorithm generated by the A-criterion is no longer superlinear. In fact, the algorithm tries to attract to the two-dimensional plane with the basis e_1, e_d by reducing the weights of the designs $\xi^{(k)}$ at the other eigenvalues. However, when it gets close to the plane, its convergence rate accelerates and the updating rule quickly recovers the weights of the other components. Then the process restarts basically at random, which creates chaos. (This phenomenon is observed in many other gradient algorithms with fast asymptotic convergence rate.)

The asymptotic rate (in the form of efficiency with respect to $R_{\min}$) of the gradient algorithm generated by the A-criterion, is shown in Fig. 1.10.

1.10 α-root algorithm and comparisons

In Fig. 1.7 we display the numerically computed asymptotic rates for the α-root gradient algorithm with $\alpha \in [0.75, 2]$, $\varrho = 50$ and $\varrho = 100$. This figure illustrates that for $\alpha < 1$ the asymptotic rate of the algorithm is $R_{\max}$. The asymptotic rate becomes much better for values of α slightly larger than 1.

Numerical simulations show that the optimal value of α depends on ϱ and on the intermediate eigenvalues. For $\varrho \leq 85$ the optimal value of α tends to be around 1.015; for $\varrho = 90 \pm 5$ the optimal value of α switches to a value of around 1.05, where it stays for larger values of ϱ.

In Fig. 1.8 we display 250 log-rates, $-\log(v(\xi_k))$, $750 < k \leq 1000$, which the α-root gradient algorithm attained starting at the random initial design ξ_0, for different α. This figure is similar to Fig. 1.4 for the steepest-descent algorithm with relaxation.

Fig. 1.9 is similar to Fig. 1.5 and shows the log-rates, $-\log(v(\xi_k))$, for the α-root gradient algorithm occurring for $\alpha \in [1.0, 1.01]$.

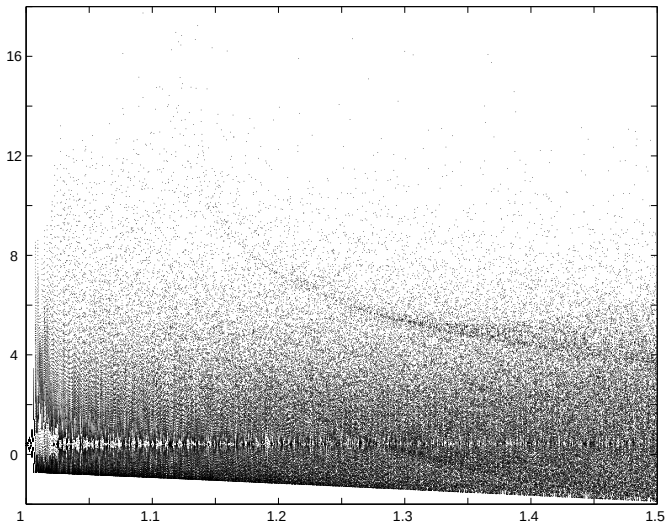

Fig. 1.8. Log-rates $-\log(v(\xi_k))$ $(750 < k \leq 1000)$ for the α-root gradient algorithm; varying α, $\varrho = 10$

In Fig. 1.10 we compare the asymptotic rates of the following gradient algorithms: (a) α-root algorithm with $\alpha = 1.05$; (b) the algorithm based on the Φ_2-optimality criterion, see (1.22); (c) steepest descent with relaxation $\varepsilon = 0.99$, see Sect. 1.7, and (d) the algorithm based on A-optimality criterion, see Sect. 1.9. The asymptotic rates are displayed in the form of efficiencies with respect to $R_{\min}$ as defined in (1.29); that is, as the ratios $R_{\min}/R$, where R is the asymptotic rate of the respective algorithm.

In Fig. 1.11 we compare the asymptotic rates (in the form of efficiencies with respect to $R_{\min}$) of the following gradient algorithms: (a) α-root algorithm with optimal value of α; (b) steepest descent with optimal value of the relaxation coefficient ε; (c) Cauchy–Barzilai–Borwein method (CBB) as defined in (Raydan and Svaiter, 2002); (d) Barzilai–Borwein method (BB) as defined in (Barzilai and Borwein, 1988).

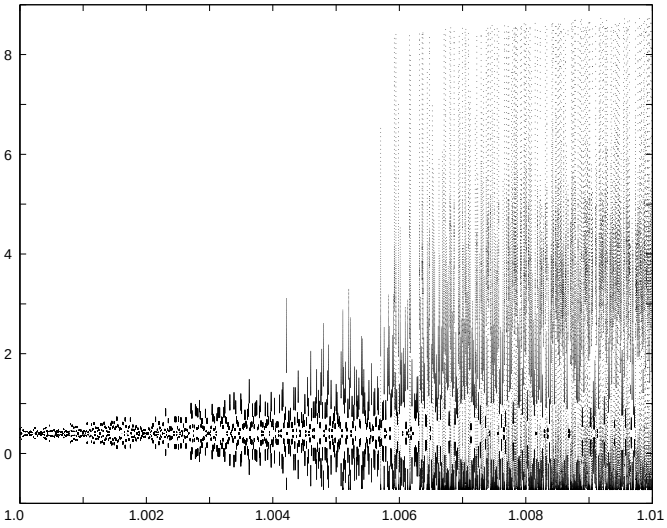

Fig. 1.9. Log-rates $-\log(v(\xi_k))$ $(750 < k \leq 1000)$ the α-root gradient algorithm; $\varrho = 10$, $\alpha \in [1.0, 1.01]$

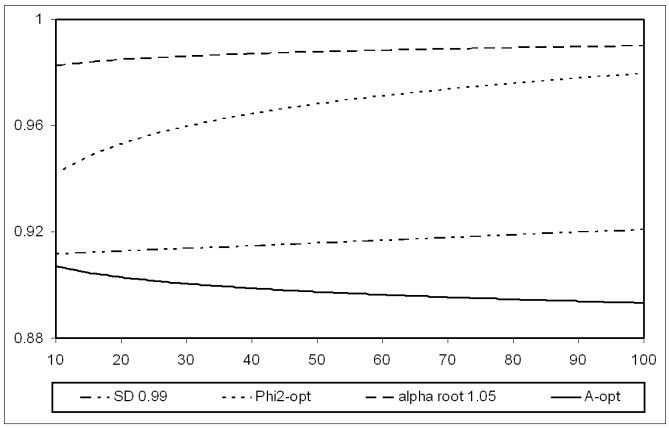

Fig. 1.10. Efficiency relative to $R_{\min}$ for various algorithms, varying ϱ

References

Akaike, H. (1959). On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Statist. Math. Tokyo*, **11**, 1–16.

Barzilai, J. and Borwein, J. (1988). Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, **8**, 141–148.

Fedorov, V. (1972). *Theory of Optimal Experiments.* Academic Press, New York.

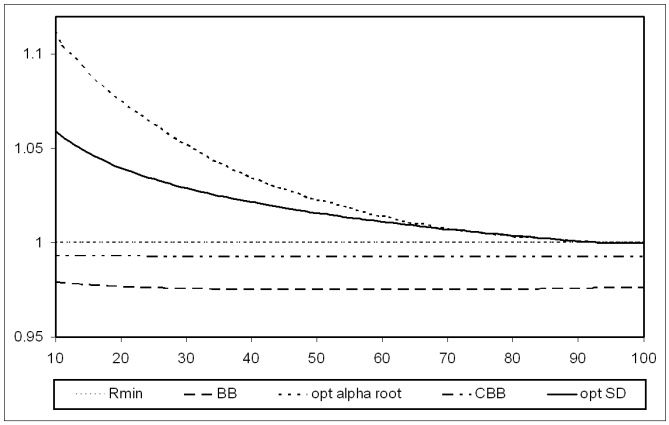

Fig. 1.11. Efficiency relative to $R_{\min}$ for various algorithms, varying ϱ (other algorithms)

Forsythe, G. (1968). On the asymptotic directions of the s-dimensional optimum gradient method. *Numerische Mathematik*, **11**, 57–76.

Kozjakin, V. and Krasnosel'skii, M. (1982). Some remarks on the method of minimal residues. *Numer. Funct. Anal. and Optimiz.*, **4**(3), 211–239.

Nocedal, J., Sartenaer, A., and Zhu, C. (2002). On the behavior of the gradient norm in the steepest descent method. *Computational Optimization and Applications*, **22**, 5–35.

Pronzato, L., Wynn, H., and Zhigljavsky, A. (2000). *Dynamical Search*. Chapman & Hall/CRC, Boca Raton.

Pronzato, L., Wynn, H., and Zhigljavsky, A. (2001). Renormalised steepest descent in Hilbert space converges to a two-point attractor. *Acta Applicandae Mathematicae*, **67**, 1–18.

Pronzato, L., Wynn, H., and Zhigljavsky, A. (2002). An introduction to dynamical search. In P. Pardalos and H. Romeijn, editors, *Handbook of Global Optimization*, volume 2, Chap. 4, pages 115–150. Kluwer, Dordrecht.

Pronzato, L., Wynn, H., and Zhigljavsky, A. (2006). Asymptotic behaviour of a family of gradient algorithms in $\mathbb{R}^d$ and Hilbert spaces. *Mathematical Programming*, **A107**, 409–438.

Raydan, M. and Svaiter, B. (2002). Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Computational Optimization and Applications*, **21**, 155–167.